METHOD FOR ENCODING/DECODING A MESSAGE AND ASSOCIATED DEVICE

The present invention relates to a method of securing and identifying messages on a network, as well as to a corresponding secure device.

A network consists of a set of sender/receiver devices suitable for exchanging messages for example via a digital bus, by radio transmission or by way of the Internet network.

To secure the flow of messages transmitted over the network between a secure sender/receiver device, commonly referred to as a certifying authority, and a client sender/receiver device, it is known to encipher the messages with the aid of enciphering keys.

In general, the device sending the messages has available an enciphering key and the receiver device a corresponding deciphering key.

The enciphering of the messages has two main types of applications:

- the securing of a message which consists of substituting an unintelligible and unutilizable text for a plaintext,

- the identification of a message which consists in guaranteeing the origin and the integrity of a message travelling over the network by using a digital signature.

In both these types of applications, it is appropriate to minimize the risks of fraudulent interception and deciphering of the messages by a third party, or of falsification by the fraudulent affixing of a signature.

Various methods of cryptography have therefore been proposed to avoid unauthorized enciphering or deciphering.

For example, so-called symmetric methods of cryptography have been proposed. In these methods, the same key, referred to as a secret key, is used for the enciphering and deciphering of a message. However, these methods are not very secure since when the secret key is discovered, all of the sender/receiver devices of the network are corrupted.

An improvement to such methods consists in using techniques referred to as derivation of symmetric keys. Figure 1 illustrates an exemplary use

of this technique. It diagrammatically represents the architecture of a certifying authority 100 and of a given client appliance 102 of a network of appliances able to communicate with this certifying authority.

According to the technique of derivation of symmetric keys, each client appliance 102 possesses its own specific enciphering/deciphering key KD_i, different from the keys of the other appliances of the network. This key is calculated or derived on the basis of an identifier CID_i stored in each client appliance 102 and of a so-called master key MK known to the certifying authority 100 alone. This derived key is used at one and the same time to encipher and to decipher a message.

The derived key KD_i is generated at the start by the certifying authority then stored in each client appliance in a secure manner. Thereafter, before each exchange of message m with a given client appliance, the certifying authority 100 requests the client appliance 102 for its identifier CID_i then recalculates the derived key KD_i of the client device concerned by applying a derivation function to the identifier CID_i and the master key MK. Next, the certifying authority enciphers (notation "E") or deciphers (notation "D") the message with the aid of the derived key calculated. The notation $E\{KD_i\}(m)$ corresponds to the enciphering of the message m with the aid of the key KD_i.

An example of so-called derivation of symmetric keys techniques used for the identification of a message is described in document WO 02/19613.

This technique is more secure than a conventional symmetric method since when a derived key of a given client appliance is hacked, not all of the client appliances of the network are corrupted since the hacker cannot calculate the derived keys of the other appliances. However, this technique is expensive since it requires the securing of all the client appliances.

Additionally, methods of asymmetric cryptography have been proposed. These methods are characterized by the use of a pair of nonidentical enciphering and deciphering keys called public key/private key.

Figure 2 illustrates an exemplary use of an asymmetric method in which a client appliance 202, 203 is able to transmit an enciphered message to a certifying authority 200.

According to this asymmetric method, each client appliance 202, 203 of the network of client appliances comprises a public key $PubC_i$, $PubC_j$ which is specific to it and which is used to encipher a message m to be transmitted. The certifying authority 200 stores in a database all the private keys corresponding to the public keys of the client appliances. The private keys are in the example of Figure 2 stored by the certifying authority 200 with the identifiers of each client appliance. When a client appliance 203 wishes to transmit an enciphered message m to the certifying authority 200, it transmits, in addition to the message m enciphered with its public key $E\{PubC_i\}(m)$, its identifier CID_i so that the certifying authority can retrieve the corresponding private key $PrivC_i$. The message m is then deciphered with the aid of the private key $PrivC_i$.

Advantageously, asymmetric methods such as these do not require the securing of the client appliances. Specifically, the hacking of a client appliance and therefore the discovery of its public enciphering key does not permit the deciphering of the message dispatched. Only the private key corresponding specifically to this public enciphering key allows the deciphering of the message.

However, the main drawback of this type of asymmetric method resides in the need for the certifying authority to manage a database in which are stored all the private keys of all the client appliances of the network. This database requires a sizable storage memory. Moreover, the search for a private key in this database involves fairly lengthy message transfer times which handicap the exchanges.

As a variant, asymmetric methods have been proposed, in which, a single pair of private/public keys enciphers all the messages. The client appliances of the network therefore all contain the same public key and the certifying authority stores a unique private key. However, these methods are not sufficiently secure since the hacking of the private key corrupts the whole of the network of client appliances.

The aim of the present invention is to provide an alternative method of enciphering/deciphering which exhibits a raised level of security without requiring the storage and the management of a database of asymmetric keys.

For this purpose, the subject of the present invention is a method of enciphering/deciphering a message to be exchanged between a sender and a

receiver by way of a communication network, the sender and the receiver both being one among a secure device and a defined client device in a network of client devices, the method comprising the steps of:

- performing operations of asymmetric cryptography by the secure device and by the defined client device respectively with the aid of a private key and of a public key, the private key being different from the public key, and

- dispatching at least one public data item from the defined client device to the secure device,

characterized in that it comprises furthermore, during each send/receive of a message enciphered by the secure device, a step of determining the private key corresponding to the public key of the defined client device, on the basis of a secret master key stored in the secure device, and the or each public data item dispatched by the defined client device.

Advantageously, this method uses the techniques of derivation of symmetric keys associated with the method of asymmetric cryptography. Thus, the derivation techniques will not be used to generate a secret derived key but to generate a private key of a pair of private/public keys.

Another subject of the invention consists of a secure device able to exchange messages with a defined client device of a network of client devices, over a communication network, the secure device being able to receive at least one public data item specific to the said defined client device and dispatched by the latter prior to any exchange of messages, the secure device comprising means for performing operations of asymmetric cryptography with the aid of a private key corresponding to a public key stored in the defined client device characterized in that it comprises, furthermore secure means of storage of a master key, and means of determination of the said private key on the basis of the master key and of the or of each public data item dispatched.

The invention will be better understood and illustrated by means of an exemplary embodiment and implementation, which are wholly nonlimiting, with reference to the appended figures, in which:

- Figure 1 is a diagrammatic view of the architecture of a certifying authority and of a receiver appliance that are able to exchange messages enciphered according to a known method of derivation of symmetric keys,

- Figure 2 is a diagrammatic view of the architecture of a certifying authority and of a sender appliance that are able to exchange messages enciphered according to a known method of asymmetric enciphering,

- Figure 3 is a diagrammatic view of the architecture of a secure device according to an exemplary embodiment of the invention for the generation of a pair of private/public keys during a phase of initialization of the appliances of the network,

- Figure 4 is a summary chart of the various steps of the method of enciphering/deciphering during the initialization phase, according to the exemplary embodiment of the invention,

- Figure 5 is a diagrammatic view of the architecture of a secure device and of a client device for the securing of a message according to the exemplary embodiment of the invention, and

- Figure 6 is a summary chart of the various steps of the method of enciphering/deciphering for the securing of a message according to the exemplary embodiment of the invention,

- Figure 7 is a diagrammatic view of the architecture of a secure device and of a client device for the identification of a message, according to an exemplary embodiment of the invention, and

- Figure 8 is a summary chart of the steps of the method of enciphering/deciphering for the identification of a message according to the exemplary embodiment of the invention.

Figure 3 diagrammatically represents the architecture of a secure device 1 and of a client device C_i.

The secure device 1 comprises a random number generator 2, a memory 3 for storing a master key, a module 4 for calculating a part d_i of the private key and a module 5 for calculating a public key $PubC_i$.

The random number generator 2 is able to generate on the one hand a number apt to constitute the so-called master key MK and on the other hand, a plurality of numbers CID_i able to identify the client devices of the network.

Preferably, the so-called master key MK has a length of 128 bits and the identifiers CID_i, CID_j of the client devices C_i, C_j have a length of 64 bits.

Additionally, the generator 2 is also able to generate at random two distinct, large odd prime numbers p and q of 512 bits used for the calculation of the public key by the calculation module 5.

The memory 3 of the secure device is nonvolatile of "ROM" or "EEPROM" type or the like. It is able to store the master key MK generated by the generator 2. As the master key is a secret key known only by the secure device, the memory 3 for storing this key is advantageously highly secure so as to guarantee the security of the messages exchanged.

The calculation module 4 is able to determine a part of a private key of a pair of private/public keys. Generally, a private key $PrivC_i$ is a mixed key consisting of two parts. The first part is formed by a part of the public key called the modulus n_i in any asymmetric algorithm. The second part is commonly called the secret exponent d_i in the asymmetric algorithms of RSA type: $PrivC_i = (n_i, d_i)$. The calculation module 4 is able to calculate the second part d_i of the private key $PrivC_i$ on the basis of the identifier CID_i of the client device C_i and of the master key MK.

The calculation module 4 preferably comprises a calculation unit 6 able to perform a function of modifying the length of an identifier CID_i into an extension of the identifier, denoted $ECID_i$. A known extension function called MGF may for example be used. This function makes it possible to extend a 64-bit number into a 1024-bit number. This function is in particular described in the document from RSA Laboratories "PKCS #1v2.1: RSA Cryptography Standard – June 14, 2002" available at the following Internet address: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-1.pdf

The calculation module 4 comprises a unit 7 for enciphering the extension of the identifier $ECID_i$ on the basis of the master key MK. This unit implements a symmetric derivation algorithm. Preferably, it entails the algorithm commonly called AES "Advanced Encryption Standard" used in CBC mode. This algorithm is described in document FIPS 197, 26 November, 2001 available on the Internet at the address: http: //csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

Advantageously, the calculation module 4 also comprises a unit 8 for selecting the secret exponent d_i as a function of the result or enciphered $ECID_i$ of

the extension of the identifier. To select this secret exponent d_i, the selection unit 8 uses a deterministic function. For example, this unit is suitable for selecting a data item such that this data item fulfils the following criteria:

- this data item d_i must be less than the result $ECID_i$ of the enciphering of the extension of the identifier,

- this data item d_i must be a number closest to the result $ECID_i$ of the enciphering of the extension of the identifier, prime to a list of prime numbers: 2, 3, 5, 7, 11, 13. Possibly, the latter condition may be extended to a longer list of prime numbers.

Diagrammatically, the determination module 5 may be decomposed into two calculation units. Each unit being able to calculate an element of the public key: $PubC_i = (n_i, e_i)$.

The first calculation unit 9 is able to select two large prime numbers p_i and q_i generated by the random number generator 2 in such a way that $(p_i-1) \times (q_i-1)$ is prime to the secret exponent d_i. In practice, a number p_i such that (p_i-1) is prime to d_i is firstly generated, followed by a number q_i such that (q_i-1) is prime to d_i.

Additionally, this calculation unit 9 is able to calculate the first part of the private key called the modulus n_i such that $n_i = p_i \times q_i$. The modulus n_i also constitutes an element of the private key $PrivC_i = (n_i, d_i)$.

The second calculation unit 10 uses an extended Euclid algorithm to calculate the other element of the public key e_i on the basis of the secret data p_i, q_i and d_i. This extended Euclid algorithm is in particular described in the work "Handbook of Applied Cryptography" by A. Menezes, P. van Oorschot and S. Vanstone, CRC Press, 1996, on page 67. This work may be consulted at the following Internet address: http://www.cacr.math.uwaterloo.ca/hac/

More precisely, we calculate the data item e_i such that:

$e_i \times d_i = 1 \bmod (p_i-1) \times (q_i-1)$.

The client devices C_i of the network comprise a memory 11 for storing an identifier CID_i and a public key $PubC_i = (n_i, e_i)$ as well as a module for asymmetric encipherment or for signature verification.

Conventionally, a secure device 1 and the client devices C_i, C_j of its communication network are personalized or initialized so as to be able to exchange enciphered messages.

The basic steps of a method of personalization of a secure device and of the client devices according to the invention will now be described.

The method of personalization according to the invention, comprises a first step of generating a unique master key MK intended for the secure device 1 and a plurality of identifiers CID_i, CID_j destined to characterize or personalize the client devices C_i, C_j of the network.

This method comprises a second step of calculating a private/public key pair associated with each client device. Specifically, the private key is obtained by enciphering the identifier CID_i of each client device C_i with the aid of the master key MK of the secure device: $PrivC_i = f\ \{MK\}\ (CID_i)$. The corresponding public key $PubC_i$ is calculated on the basis of the private key in particular by applying a mathematical function using for example an extended Euclid algorithm: $PubC_i = F\ (PrivC_i)$.

According to a third step of the method of personalization of the secure device and of the client devices of the network, the identifiers CID_i, CID_j generated and the public keys $PubC_i$, $PubC_j$ calculated on the basis of said identifiers are dispatched to each client device C_i, C_j of the network or are inserted into the client devices during their manufacture.

Finally, the corresponding private keys $PrivC_i$, $PrivC_j$, as well as the whole set of intermediate data that make it possible to calculate the private/public key pairs are destroyed. Thus, the secure device stores no data item associated with any one of these client devices.

The steps of an exemplary embodiment of the method of personalization will now be described in conjunction with Figure 4.

During a step 41 of the phase of personalization of the devices of the network, the generator 2 generates a random number of 128 bits which constitutes the master key MK and a number of 64 bits which is able to become the identifier CID_i of a client device C_i to be personalized.

During a step 42, the master key MK thus generated is stored in the memory 3 of the secure device 1. This master key MK will serve as basis for the

calculation of the whole set of private/public key pairs associated with all the client devices of the network.

During a step 43, the calculation unit 6 extends the identifier CID_i of a client device C_i via an extension algorithm so as to generate a 128-bit number forming the extension of the identifier $ECID_i$.

The extension of the identifier $ECID_i$ is then enciphered in step 44 with the aid of the master key MK. This enciphering is carried out by the calculation unit 7 by applying a symmetric algorithm of AES type.

Next, during a step 45, the selection unit 8 selects a number forming the secret exponent d_i.

In the course of steps 46 and 47, the calculation module 5 selects two large prime numbers p_i and q_i and calculates the public key $PubC_i = (n_i, e_i)$ on the basis of these numbers and of the secret exponent d_i.

Once the public key $PubC_i = (n_i, e_i)$ of a given client device C_i has been calculated, the secure device 1 dispatches the former to the latter in a safe manner, not detailed here, accompanied by the identifier CID_i from which the calculation of this public key originates in step 48.

The identifier CID_i and the public key $PubC_i$ are recorded in the memory 11 of the client device C_i.

Advantageously, according to the invention, the memory 11 of the client devices need not be made secure against reading since the discovery of the public key $PubC_i$ and of the identifier CID_i does not in any way allow the calculation of the corresponding private key $PrivC_i$ or the calculation of another private or public key of the network, so that the security of the enciphered message transmitted and of the network of sender/receiver devices is preserved.

Furthermore, the identifier CID_i as well as the whole set of data calculated on the basis thereof and in particular the secret data p_i and q_i, the secret exponent d_i, the public exponent e_i, the modulus n_i, and the extension of the identifier $ECID_i$ are not retained in the memory 3 of the secure device 1 and are destroyed in step 49.

Consequently, the hacking of the master key MK does not allow the calculation of the private/public keys associated with a given client device without knowing its identifier.

The method of personalization is aimed at configuring the secure device and the client devices in such a way as to allow the exchanging of the messages enciphered with a view to their securing or to their identification.

An exemplary use of the sender/receiver devices according to the invention with reference to Figures 5 and 6 will now be described.

In particular, Figure 5 represents the architecture of a given client device C_j able to dispatch an enciphered message E {Pub C_j} (m) as well as the architecture of a secure device 1 able to decipher this message.

Conventionally, the client device C_i comprises a non-volatile memory 11 and an enciphering module 12.

The memory 11 of the client device C_j comprises an identifier CID_j and a public key $PubC_j$ composed of a modulus n_j, and of a public data item e_j.

The secure device 1 comprises a memory 3 in which the master key MK is stored, a module 4 for calculating the secret exponent d_j and a deciphering module 13.

According to the invention, the enciphering module 12 and the deciphering module 13 use methods of asymmetric cryptography implementing algorithms such as for example the algorithm RSAES-OAEP. A description of this algorithm may be found in the document « PKCS #1v2.1: #RSA Cryptography Standard » which has already been mentioned previously.

The module 4 for calculating the secret exponent d_j comprises the same calculation units as the calculation module 4 used during the phase of personalization of the client devices. Consequently, it calculates the secret exponent d_i on the basis of the identifier CID_i of the client device C_i and of the master key MK in the same way as during the personalization phase so that this secret exponent d_i still corresponds to the public enciphering key $PubC_i$ stored in the memory 11 of the client device C_i.

The method of enciphering/deciphering for securing a message will be described in detail in conjunction with Figure 6.

This method comprises a step 61 of enciphering the message to be transmitted. This enciphering is carried out by the enciphering module 12 of the client device C_j with the aid of the public key $PubC_j = (n_j, e_j)$.

E {Pub C_j} (m) = RSAES-OAEP Encrypt {(n_j, e_j)} (m)

Next, during a step 62, the identifier CID_j and the modulus n_j of the client device C_j as well as the enciphered message E {Pub C_j } (m) are dispatched to the secure device 1.

Finally, the units for calculation 6, 7 and for selection 8 of the module 4 for calculation of the secret exponent d_j of the secure device 1 carry out a step 63 of calculation of the extension of the identifier $ECID_j$ on the basis of the identifier CID_j dispatched by the client device C_j, a step of enciphering 64 of the extension of the identifier $ECID_j$ with the aid of the master key MK and a step of selection 65 of the secret exponent d_j on the basis of the result $ECID_i$ of the enciphering of the extension of the identifier. It is necessary that the selection unit 8 use the same selection rules as those applied during the phase of personalization of the client devices.

Lastly, the module for asymmetric deciphering 13 of the secure device 1 carries out a step 66 of deciphering the message with the aid of the mixed private key composed of the calculated secret exponent d_j and of the modulus n_j which is dispatched by the client device C_j:

$$m = \text{RSAES - OAEP} - \text{Decrypt } \{(d_j, n_j)\} (E \{\text{Pub } C_j\} (m))$$

Advantageously the secure device 1 retains no data item tied to the client device C_j sending a message. Specifically, its identifier CID_j, the extension $ECID_j$ of its identifier, its secret exponent d_j and its modulus n_j are destroyed during step 67.

The enciphering/deciphering method of the invention also makes it possible to identify a message by affixation of a signature by the secure device 1 and verification of this signature by a client device C_j for which the signed message is intended.

The inventive enciphering/deciphering method used to identify the origin of a message will be described in conjunction with Figures 7 and 8.

Figure 7 diagrammatically represents the architecture of a secure device 1 and of a client device C_j.

The system composed of a secure device and of a client device is similar to the system described in conjunction with Figure 5. Consequently, the elements common to Figures 5 and 7 employ the same references and will not be described again.

In fact, the secure device/client device system comprises the same modules 4 and memories 3, 11 apart from the module for deciphering 13 of the secure device and the module for enciphering 12 of the client device which are replaced respectively with a signature generation module 14 and with a signature verification module 15.

The enciphering/deciphering method used for the signing of a message comprises a step 81 in the course of which the secure device 1 requests the identifier CID_j and the modulus n_j from the client device C_j to which it wishes to dispatch a signed message m.

In the course of steps 82, 83 and 84, the module for calculation 4 of the secure device 1 recalculates the secret exponent d_j of the client device C_j on the basis of the identifier dispatched CID_j and of the master key MK in the same manner as in the enciphering/deciphering method used for the securing or the personalizing of a message m described previously.

Next, during a step 85, the module for signature 14 of the secure device 1 signs its message with the aid of the secret exponent d_j calculated and of the modulus n_j dispatched by the client device C_j: $S\{PrivC_j\}(m)$ with $PrivC_j = (d_j, n_j)$.

Finally, in the course of step 86, the securing device 1 dispatches a message m as well as its signature $S\{(d_j, n_j)\}(m)$ to the defined client device C_j.

During a step 87, the module for verification 15 of the client device C_j verifies the signature of the message with the aid of the public key $PubC_j = (n_j, e_j)$ stored in its memory 11 and corresponding to the private key $PrivC_j = (d_j, n_j)$ by performing the operation:

$V\{PubC_j\}(S\{PrivC_j\}(m)) = 0$ or 1

During a step 88, the identifier CiD_j of the defined client device C_j and the intermediate data CID_j, $ECID_j$, d_j and n_j that made it possible to determine the private key are destroyed by the secure device.

For the signature operation S and signature verification operation V, it will in particular be possible to use the RSASSA-PSS algorithm which is described in the document « PKCS#1v2.1:RSA Cryptography Standard » mentioned above.